PGR112 – Step 13: Intro to JDBC

Object Oriented Programming

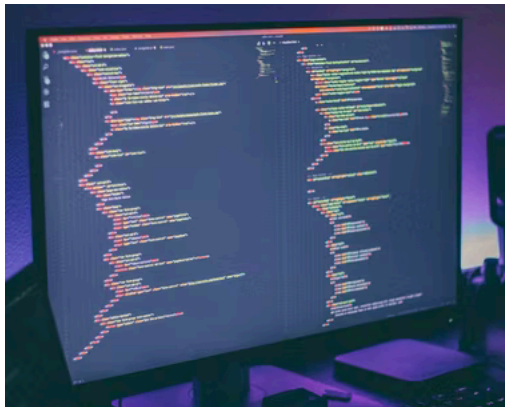
Bogdan Marculescu / bogdan.marculescu@kristiania.no

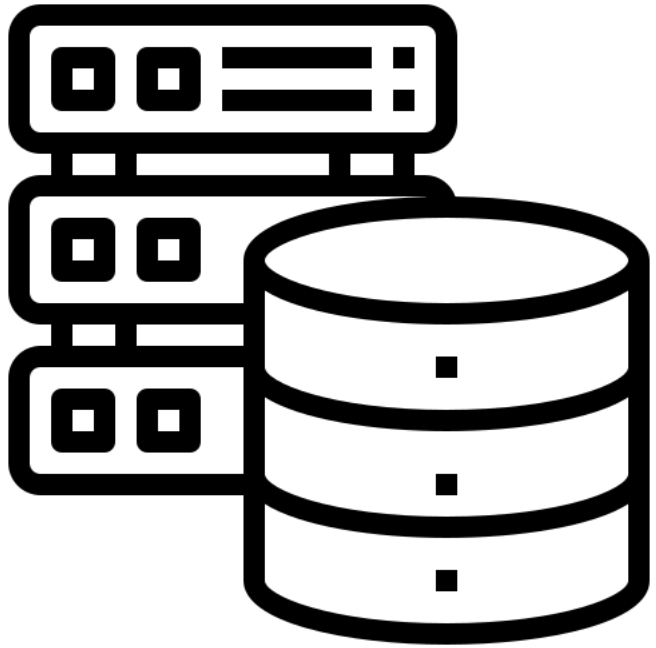
Agenda

- What is JDBC?
- Why use it?
- How do we use it?

Using more than text files

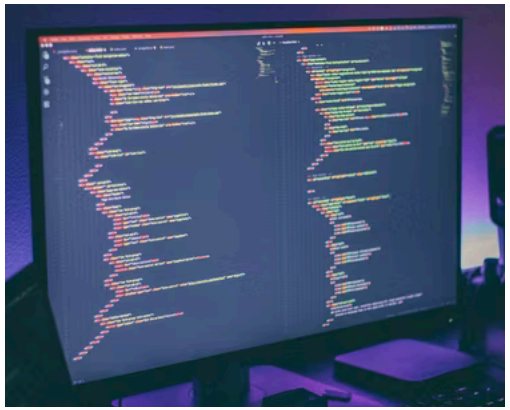
- You already know a thing or two about databases
- We could access the database directly, right?

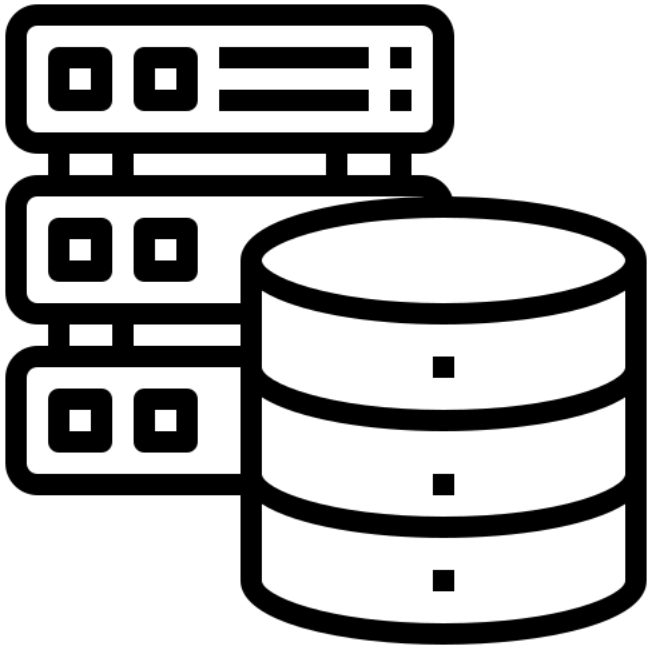




Database connections

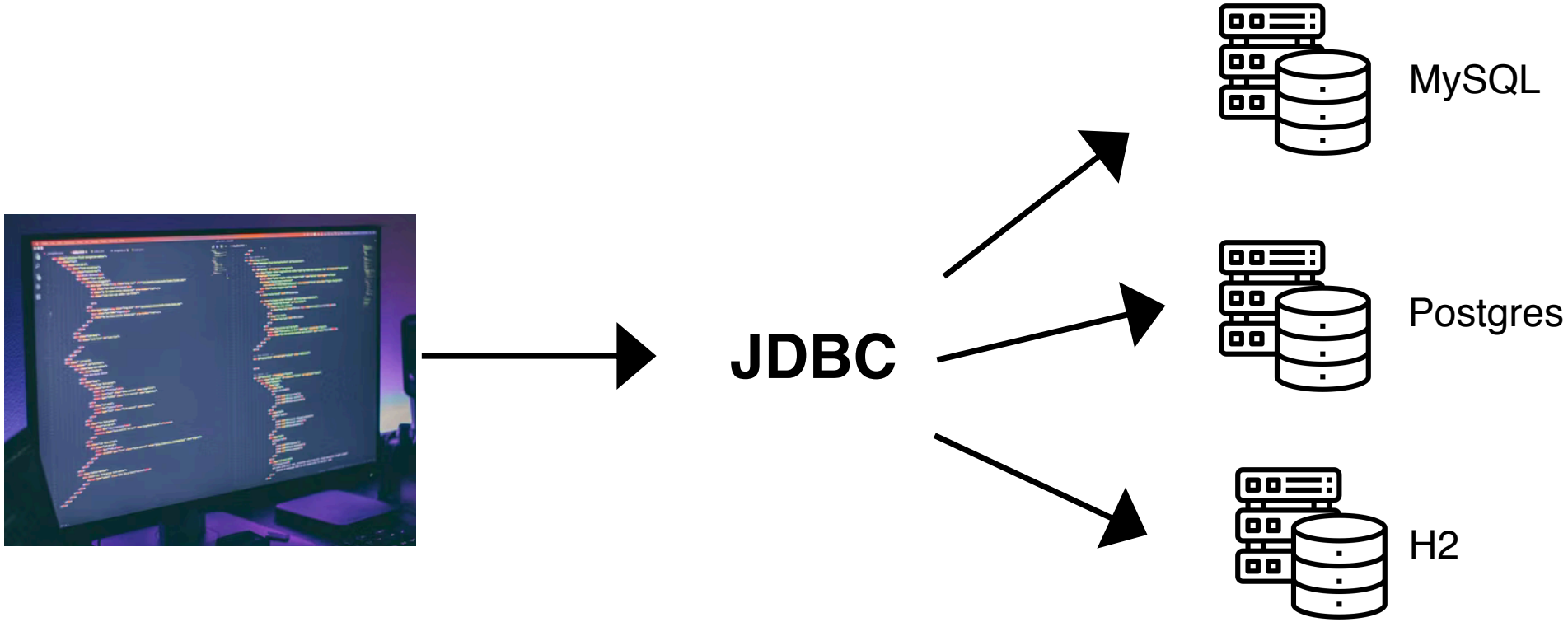
- What if the DB provider changes technology?
- Or we want to change DB providers?





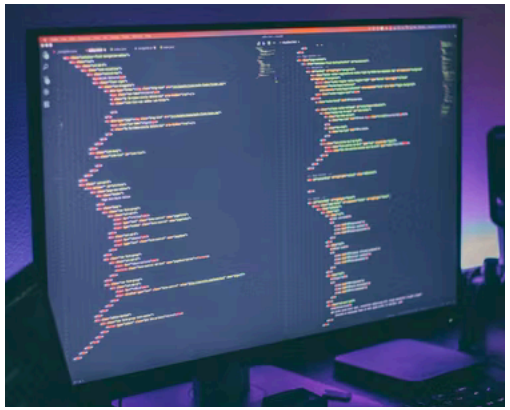
Database connections

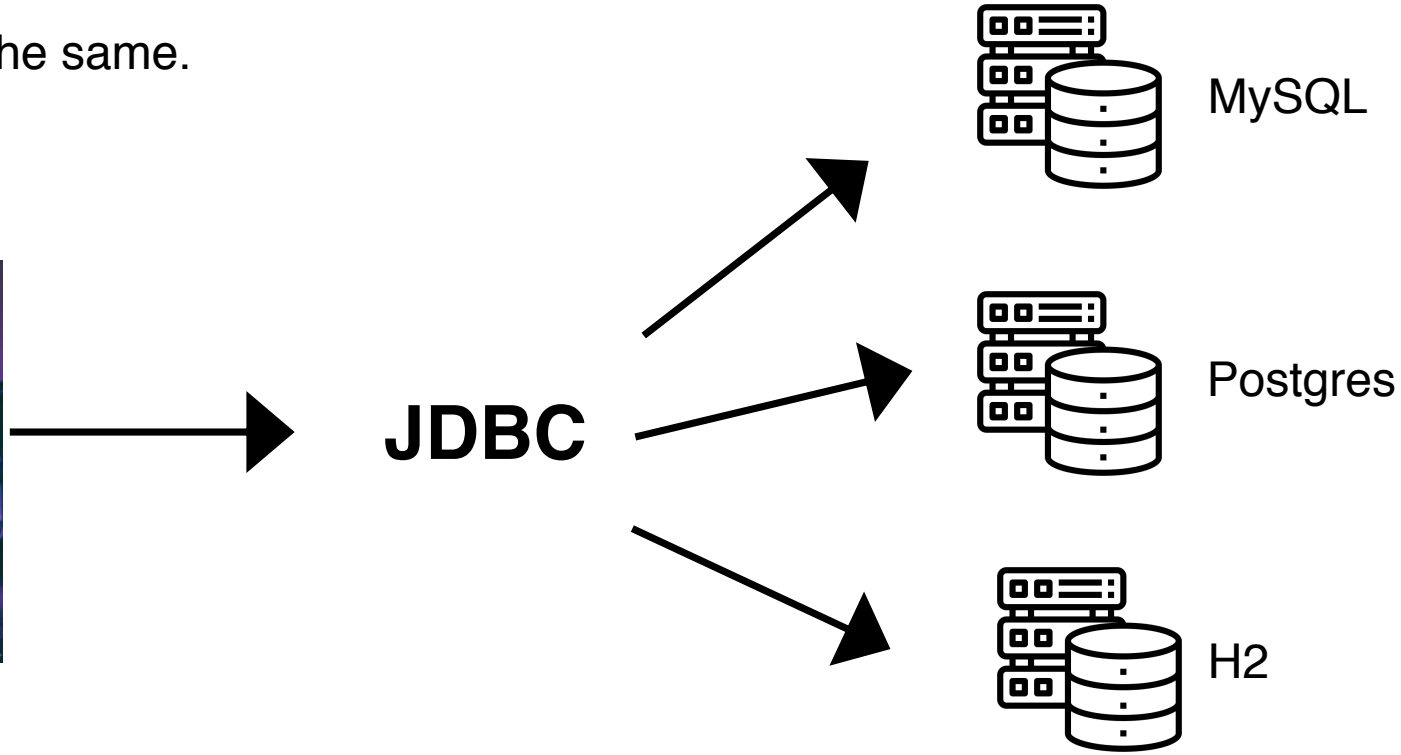
- Instead



Database connections

- Any changes are handled in the Drivers
- Our code can remain the same.





First

- Make sure you have MySQL installed (you can use any DB you like, this example uses that)
- Ensure you have the right permissions, access, users, etc.
- Create the DB you intend to use
  - Quick reminder:
  - CREATE DATABASE equipmentBallsDb
  - SHOW DATABASES

```
mysql> CREATE DATABASE equipmentBallsDb
    -> ;
Query OK, 1 row affected (0.09 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| equipmentBallsDb   |
| information_schema |
| myShapesDb         |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.00 sec)
```

Some things can be stored as scripts

- In resources/lesson13/sql – you can find some sql scripts (e.g. for setting up and dropping tables).
- This is fine, but…
- We want to programmatically add objects
  - E.g. objects that result from other computations

```sql
USE equipmentBallsDb;

CREATE TABLE lockers(
    id INT NOT NULL AUTO_INCREMENT,
    location VARCHAR(45),
    primary key (id)
);

CREATE TABLE equipmentBall(
    id INT NOT NULL AUTO_INCREMENT,
    type VARCHAR(45) NOT NULL ,
    needsAir BOOLEAN,
    location INT NULL,
    primary key(id),
    CONSTRAINT fk_location
        FOREIGN KEY (location)
        REFERENCES lockers(id)
);
```

Basics of interaction with a database

- Adding a JDBC connect
- Add the MySQL j-Connector jar. (File -> Project Structure ->Libraries->From Maven)

Basics of interaction with a database

- java.sql.Connection – handling a connection to a database
- java.sql.DriverManager – the means of getting a Connection object (in java, everything is an object, remember).
- We use the Connection object to create a Statement:

```
try (Connection con = DriverManager
        .getConnection( url: "jdbc:mysql://localhost:3306/equipmentBallsDb?useSSL=false",
                user: "root",
```

```
Statement stmt = con.createStatement();
```

Creating a query programmatically

- You know about queries:
  - INSERT INTO persons (name, address) VALUES ('John', 1);

- But…
  - What if we want to insert a particular object?

```
int needsAir = (ball.needsAir()) ? 1 : 0;

String insertSql = "INSERT INTO equipmentBall(type, needsAir, location)"
        + " VALUES('" +
        ball.getType() + "', '" +
        needsAir + "', '" +
        ball.getLocation().getId() +
        "')";

stmt.executeUpdate(insertSql);
```

Creating a query programmatically

- You know about queries:
  - SELECT * FROM …;

```
String selectSql = "SELECT * FROM lockers";

ResultSet resultSet = stmt.executeQuery(selectSql);
while (resultSet.next()){
    System.out.println(resultSet.getString( columnLabel: "location"));
    Locker lock = new Locker();
    lock.setId(resultSet.getInt( columnLabel: "id"));
    lock.setLocation(resultSet.getString( columnLabel: "location"));
    results.add(lock);
}
```

Creating a query programmatically

- You know about queries:
  - SELECT * FROM …;

```
String selectLocker = "SELECT * FROM lockers " +
        "WHERE id='" +
        id +
        "'" ;

ResultSet resultSet = stmt.executeQuery(selectLocker);
while(resultSet.next()){
    Locker l1 = new Locker();
    l1.setId(resultSet.getInt( columnLabel: "id"));
    l1.setLocation(resultSet.getString( columnLabel: "location"));
     return l1;
}
```

Conclusion

- JDBC is an API that allows a program to connect to a DB
- JDBC allows the program itself to be agnostic of the underlying technology used, so long as the drivers are okay.
- You now know:
  - How to create a DB Connection, and use it to interact with a DB
  - How to create a query to insert objects created during program execution
  - How to retrieve all or some objects during program execution.

More material (tutorials and such)

- https://www.javatpoint.com/java-jdbc
- https://docs.oracle.com/javase/tutorial/jdbc/basics/gettingstarted.html
- https://www.baeldung.com/java-jdbc
- https://alvinalexander.com/java/edu/JDBC-SQLProcessor/Simple_JDBC_Example.shtml
- https://www.tutorialspoint.com/jdbc/jdbc-sample-code.htm